49

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/019,828	05/21/2002	Chenghui Luo	Fraunh01.013	8663

25247 7590 12/19/2006

GORDON E NELSON
PATENT ATTORNEY, PC
57 CENTRAL ST
PO BOX 782
ROWLEY, MA 01969

EXAMINER	
BAUM, RONALD	
ART UNIT	PAPER NUMBER
2136	

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	12/19/2006	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

Office Action Summary	**Application No.** 10/019,828	**Applicant(s)** LUO ET AL.	
	Examiner Ronald Baum	**Art Unit** 2136	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *20 October 2006*.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-13* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-13* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) □ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application (PTO-152)

6) □ Other: _____.

DETAILED ACTION

1. This action is in reply to applicant's correspondence of 20 October 2006.

2. Claims 1-13 are pending for examination.

3. Claims 1-13 are rejected.

Claim Objections

Claims 6-8 are objected to under 37 CFR 1.75(c) as being in improper form because a multiple dependent claim 3-5. See MPEP § 608.01(n). For the sake of applying art, the examiner assumes the claims depend on claim 5. Correction is required.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

4. Claims 1-13 are rejected under 35 U.S.C. 102(e) as being anticipated by Collberg et al, U.S. Patent No. 6,668,325 B1.

5. As per claim 1; "A method of obfuscating executable code that uses a first reference including a symbolic object name and a symbolic field name to reference a field containing data the method comprising the steps of:

defining an object

wherein the field is not referenced by

a symbolic field name [figures 4-7,11-19,24-32 and associated descriptions, whereas the JAVA object instances/class/code transformations used for the obfuscation are instantiated and therefore defined, whereas the object transformations are inclusive of the inherent referencing characteristics of objects inclusive of arrays of objects (and associated references) as they are used/referenced during execution, etc., clearly encompasses the claimed limitations, as broadly interpreted by the examiner.]; and

replacing the first reference with a second reference that

references the field by

the defined object's name and

the field

as required by the defined object [figures 4-7,11-19,24-32 and associated descriptions, whereas the JAVA object instances/class/code transformations used for the obfuscation, inclusive of substitution transformations, clearly encompasses the claimed limitations, as broadly interpreted by the examiner.].".

6. As per claim 2; "A method of obfuscating executable code in a language that includes classes and methods that permit reflection, the method comprising the steps of:

using the classes and methods that permit reflection to produce

one or more first constructs that have the same effect as

a second construct in the executable code that

does not employ reflection [figures 4-7,11-19,24-32 and associated descriptions, whereas the JAVA object instances/class/code transformations used for the obfuscation are instantiated and therefore defined, whereas the object transformations are inclusive of the inherent referencing characteristics of objects inclusive of arrays of objects (and associated references), classes and methods permissive of standard features (i.e., reflection) so as to allow constructs aliasing/substitution as they are used/referenced during execution, etc., clearly encompasses the claimed limitations, as broadly interpreted by the examiner.]; and

replacing

the second construct with

the one or more first constructs [figures 4-7,11-19,24-32 and associated descriptions, whereas the JAVA object instances/class/code constructs/transformations used for the obfuscation, inclusive of constructs aliasing/substitution transformations, clearly encompasses the claimed limitations, as broadly interpreted by the examiner.].".

7. As per claim 3; "A method of executing obfuscated code that includes a portion that relates a first construct whose definition is local to the executable code to a second construct whose definition is external to the executable code and that has been obfuscated by encrypting at least the second construct the method comprising the steps of:

receiving code that includes the portion [figures 4-7,11-19,24-32 and associated descriptions, whereas the JAVA object instances/class/code transformations used for the obfuscation are instantiated and therefore defined, whereas the said JAVA object instances/class/code transformations, both local and network (i.e., remote accessible/referenced objects), so as to allow constructs aliasing/substitution as they are used/referenced during execution, etc., clearly encompasses the claimed limitations, as broadly interpreted by the examiner.]; and

when the executable code is executed,

employing a key and cryptographic apparatus to relate

the second construct to

the external definition therefor [figures 4-7,11-19,24-32 and associated descriptions, whereas the JAVA object instances/class/code transformations used for the obfuscation are instantiated and therefore defined, whereas the said JAVA object instances/class/code transformations (i.e., inclusive of encryption cryptographic transformations specifically) inherent referencing characteristics, both local and network (i.e., remote accessible/referenced objects), so as to allow constructs aliasing/substitution as they are used/referenced during execution, etc., clearly encompasses the claimed limitations, as broadly interpreted by the examiner.].".

8. Claim 4 ***additionally recites*** the limitations that; "The method of executing obfuscated code set forth in claim 3 wherein the step of employing the cryptographic apparatus includes the steps of:

using a decryption key with the cryptographic apparatus to decrypt
the encrypted second construct; and
using the decrypted second construct to relate
the first construct to
the external definition.".

The teachings of Collberg et al (figures 4-7,11-19,24-32 and associated descriptions, whereas the JAVA object instances/class/code transformations used for the obfuscation are instantiated and therefore defined, whereas the said JAVA object instances/class/code transformations (i.e., inclusive of encryption cryptographic transformations specifically, of which associated decryption via appropriate key information would clearly have to occur), so as to allow constructs aliasing/substitution as they are used/referenced during execution, etc., clearly encompasses the claimed limitations, as broadly interpreted by the examiner.) suggest such limitations.

9. Claim 5 ***additionally recites*** the limitations that; "The method of executing obfuscated code set forth in claim 3 wherein the step of employing the cryptographic apparatus includes the steps of:

using an encryption key with the cryptographic apparatus to encrypt
at least the second construct in the external definition; and
using the encrypted second construct from the external definition to relate
the encrypted second construct from the executable code to
the external definition,
whereby the first construct is related to
the external definition.".

The teachings of Collberg et al (figures 4-7,11-19,24-32 and associated descriptions, whereas the JAVA object instances/class/code transformations used for the obfuscation are instantiated and therefore defined, whereas the said JAVA object instances/class/code transformations (i.e., inclusive of encryption cryptographic transformations specifically, of which associated decryption via appropriate key information would clearly have to occur), so as to allow constructs aliasing/substitution as they are used/referenced during execution, etc., clearly encompasses the claimed limitations, as broadly interpreted by the examiner.) suggest such limitations.

10. Claim 6 ***additionally recites*** the limitations that; "The method of executing obfuscated code set forth in any one of claims 3, 4 or 5 wherein:

the executable code includes

a plurality of the first and second constructs contained in

a plurality of the portions; and

a plurality of keys and the cryptographic apparatus are employed to relate

the second constructs to

the external definitions therefor.".

The teachings of Collberg et al (figures 4-7,11-19,24-32 and associated descriptions, whereas the JAVA object instances/class/code transformations used for the obfuscation are instantiated and therefore defined, whereas the said JAVA object instances/class/code transformations (i.e., inclusive of encryption cryptographic transformations specifically, of which associated decryption via appropriate key information would clearly have to occur), so as to allow constructs aliasing/substitution as they are used/referenced during execution, etc., clearly encompasses the claimed limitations, as broadly interpreted by the examiner.) suggest such limitations.

11. Claim 7 ***additionally recites*** the limitations that; "The method of executing obfuscated code set forth in any one of claims 3, 4 or 5 wherein:

the second constructs are

class specifiers; and

the step of employing a key and cryptographic apparatus is performed in

a loader for the class specifiers.".

The teachings of Collberg et al (figures 4-7,11-19,24-32 and associated descriptions, whereas the JAVA object instances/class/code transformations used for the obfuscation are instantiated and therefore defined, whereas the said JAVA object instances/class/code transformations (i.e., inclusive of encryption cryptographic transformations specifically, of which associated decryption via appropriate key information would clearly have to occur), so as to allow constructs aliasing/substitution as they are used/referenced during class loading/object instantiation (inclusive of aspects of the class specifiers)/execution, etc., clearly encompasses the claimed limitations, as broadly interpreted by the examiner.) suggest such limitations.

12. Claim 8 ***additionally recites*** the limitations that; "The method of executing obfuscated code set forth in any one of claims 3, 4 or 5 wherein:

in the step of receiving,

the code is downloaded ; and

the step of employing a key and cryptographic apparatus is performed

after downloading.".

The teachings of Collberg et al (figures 4-7,11-19,24-32 and associated descriptions, whereas the JAVA object instances/class/code transformations used for the obfuscation are instantiated and therefore defined, whereas the said JAVA object instances/class/code transformations, both local and network (i.e., remote accessible/referenced/downloaded objects and encryption cryptographic information where appropriate), so as to allow constructs aliasing/substitution as they are used/referenced during execution, etc., clearly encompasses the claimed limitations, as broadly interpreted by the examiner.) suggest such limitations.

13. As per claim 9; "A method of obfuscating executable code that includes a portion that relates a first construct whose definition is local to the executable code to a second construct whose definition is external to the executable code, the method comprising the steps of:

locating the portion; and

encrypting at least the second construct [figures 4-7,11-19,24-32 and associated descriptions, whereas the JAVA object instances/class/code transformations used for the obfuscation are instantiated and therefore defined, whereas the said JAVA object instances/class/code transformations, both local and network (i.e., remote accessible/referenced/downloaded objects and encryption cryptographic information where appropriate), so as to allow constructs aliasing/substitution as they are used/referenced during execution, etc., clearly encompasses the claimed limitations, as broadly interpreted by the examiner.].".

14. Claim 10 ***additionally recites*** the limitations that; "The method of obfuscating executable code set forth in claim 9 wherein

there are a plurality of first and second constructs contained in

a plurality of the portions; and

in the step of encrypting at least the second construct

a plurality of keys is employed to encrypt the second constructs in

the plurality of portions.".

The teachings of Collberg et al (figures 4-7,11-19,24-32 and associated descriptions, whereas the JAVA object instances/class/code transformations used for the obfuscation are instantiated and therefore defined, whereas the said JAVA object instances/class/code transformations, both local and network (i.e., remote accessible/referenced/downloaded objects and encryption cryptographic information where appropriate, further inclusive of the said cryptographic information granularity encompassing the information as a plurality of cryptographic information), so as to allow constructs aliasing/substitution as they are used/referenced during execution, etc., clearly encompasses the claimed limitations, as broadly interpreted by the examiner.) suggest such limitations.

15. As per claim 11; "A method of executing a construct that is encrypted in executable code without decrypting the encrypted construct the construct being one of a plurality of constructs belonging to an execution environment in which the executable code will execute and the method comprising the steps of:

using an encryption key that was used to encrypt the construct in the executable code to encrypt the constructs in the execution environment [figures 4-7,11-19,24-32 and associated descriptions, whereas the JAVA object instances/class/code transformations used for the obfuscation and subsequent execution are instantiated and therefore defined, whereas the said JAVA object instances/class/code transformations, both local and network (i.e., remote/mobile accessible/referenced/downloaded objects and encryption cryptographic information where appropriate), so as to allow constructs aliasing/substitution as they are used/referenced during execution within the appropriate run time environment (i.e., JRE, etc.,), clearly encompasses the claimed limitations, as broadly interpreted by the examiner.];

comparing

the encrypted construct in the executable code with

the encrypted constructs in the execution environment; and

when a match is found, executing the encrypted construct in the executable code using

the unencrypted construct in the execution environment that corresponds to

the matching encrypted construct in the execution environment [figures 4-7,11-19,24-32 and associated descriptions, whereas the JAVA object instances/class/code transformations used for the obfuscation and subsequent execution are instantiated and therefore defined, whereas the said JAVA object instances/class/code transformations, both local and network (i.e., remote/mobile accessible/referenced/downloaded objects and encryption cryptographic information where appropriate), so as to allow constructs aliasing/substitution as they are used/referenced during execution within the appropriate run time environment (i.e., JRE, etc.,) such that said environment would only be used for usable (i.e., cryptographically unencrypted or subsequently decrypted), clearly encompasses the claimed limitations, as broadly interpreted by the examiner.].".

16. Claim 12 ***additionally recites*** the limitations that; "The method of executing a construct set forth in claim 11 wherein:

the executable code is

mobile code; and

the steps of the method are performed in an apparatus to which the mobile code has been downloaded.".

The teachings of Collberg et al (figures 4-7,11-19,24-32 and associated descriptions, whereas the JAVA object instances/class/code transformations used for the obfuscation and subsequent execution are instantiated and therefore defined, whereas the said JAVA object instances/class/code transformations, both local and network (i.e., remote/mobile accessible/referenced/downloaded objects and encryption cryptographic information where appropriate), so as to allow constructs aliasing/substitution as they are used/referenced during execution, etc., clearly encompasses the claimed limitations, as broadly interpreted by the examiner.) suggest such limitations.

17. As per claim 13, this claim is the apparatus for the method claim 1 above, and is rejected for the same reasons provided for the claim 1 rejection; "A data storage device for use with a computer, the data storage device being characterized in that:

the data storage device contains code which, when executed by the computer, causes the computer to perform the method set forth in any one of claims 1, 2, 3, 9, or 11.".

Response to Amendment

18. As per applicant's argument concerning the lack of teaching by the cited reference, the examiner agrees because the examiner cited the incorrect reference patent (the patent was also not cited in the PTO-892), but has been corrected in this non-final office action.

Conclusion

19. Any inquiry concerning this communication or earlier communications from examiner should be directed to Ronald Baum, whose telephone number is (571) 272-3861, and whose unofficial Fax number is (571) 273-3861 and unofficial email is Ronald.baum@uspto.gov. The examiner can normally be reached Monday through Thursday from 8:00 AM to 5:30 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Nasser Moazzami, can be reached at (571) 272-4195. The Fax number for the organization where this application is assigned is **571-273-8300**.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. For more information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

NASSER MOAZZAMI
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

12/12/06

Ronald Baum

Patent Examiner